

Phân Tích Dữ Liệu Thực Tế với Python

Bài 11.2: Sử Dụng SQL cho EDA



Quang-Khai Tran, Ph.D
CyberLab, 04/2023



(Ảnh: Internet)

Nội dung



1. Một số mệnh đề SQL nâng cao
2. Kết hợp SQL và Pandas
3. Mở rộng:
 - (1) Kết nối DBMS với Python
 - (2) NoSQL là gì?
4. Bài tập & Thảo Luận

1	SELECT	
2	JOIN	INNER JOIN, LEFT JOIN, RIGHT JOIN, OUTER JOIN
3	WHERE	
4	GROUP BY	
5	HAVING	
6	Window Function	ROW_NUMBER(), SUM(), RANK(), AVG()
7	UNION	
8	CREATE	
9	INSERT	
10	UPDATE	
11	DELETE	
12	DROP	DROP TABLE, DROP INDEX, DROP VIEW, DROP PROCEDURE
13	ALTER	ALTER TABLE, ALTER INDEX, ALTER VIEW



Tham khảo: 13 lệnh SQL cho 90%

<https://levelup.gitconnected.com/13-sql-statements-for-90-of-your-data-science-tasks-27902996dc2b>



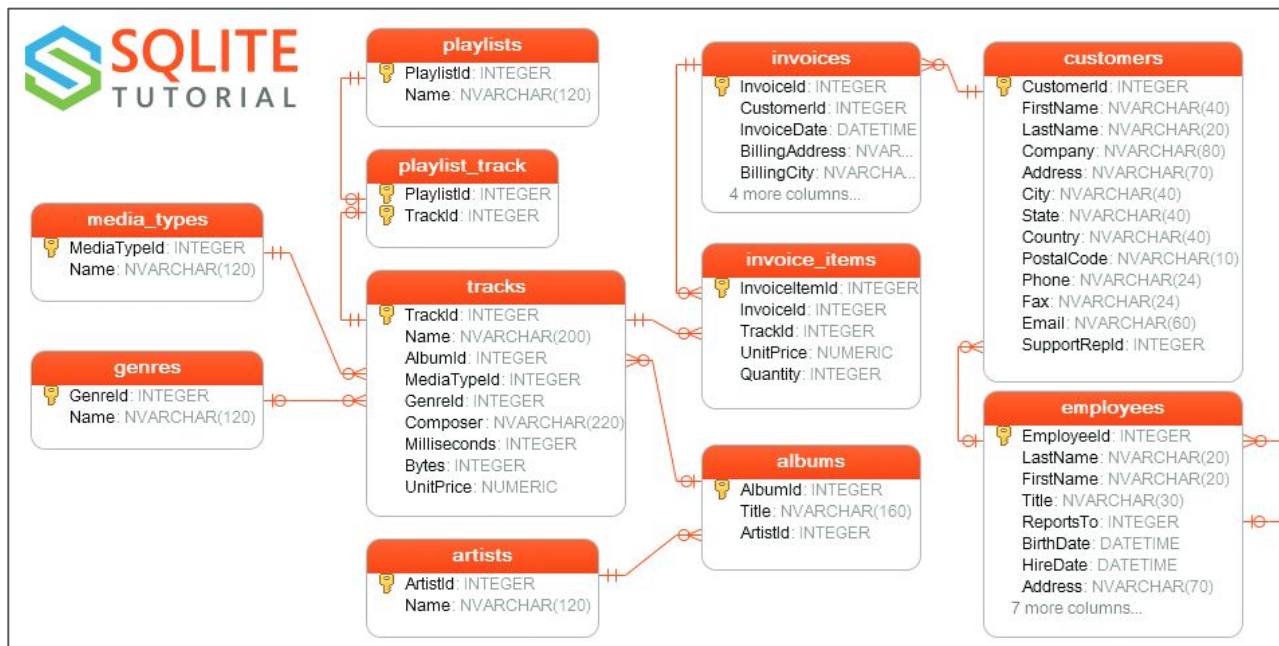
Phần 1. Truy vấn SQL nâng cao

Structured Query Language



- 1.1. Gom nhóm dữ liệu**
(GROUP BY/HAVING)
- 1.2. Truy vấn dữ liệu từ nhiều bảng**
(Các mệnh đề JOIN, UNION)

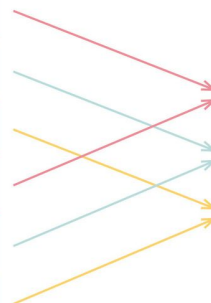
- ❖ Tham khảo: <https://www.sqlitetutorial.net/>
- ❖ Ví dụ: Chinook database (Link: <https://www.sqlitetutorial.net/sqlite-sample-database/>)



1.1 Gom nhóm dữ liệu

Các lệnh/mệnh-đề (clause) gom nhóm

- ❖ Thực hiện gom nhóm: GROUP BY
- ❖ Là mệnh đề tùy chọn sau lệnh SELECT, và phải xếp sau FROM, WHERE
- ❖ Kết hợp các hàm tính toán: MAX, MIN, SUM, COUNT, AVG
- ❖ Kết hợp điều kiện sau gom nhóm: HAVING
- ❖ Lưu ý: kết hợp điều kiện cho SELECT là mệnh đề WHERE



title	genre	price
book 1	adventure	11.90
book 2	fantasy	8.49
book 3	romance	9.99
book 4	adventure	9.99
book 5	fantasy	7.99
book 6	romance	5.88

genre	avg_price
adventure	$(11.90 + 9.99)/2$ 10.945
fantasy	$(8.49 + 7.99)/2$ 8.24
romance	$(9.99 + 5.88)/2$ 7.935

1.2 Truy vấn dữ liệu từ nhiều bảng

Sử dụng các mệnh đề JOIN

JOIN	lấy ra các dòng xuất hiện ở cả 2 bảng
INNER JOIN	lấy ra các dòng xuất hiện ở cả 2 bảng
CROSS JOIN	lấy ra các dòng xuất hiện ở cả 2 bảng
LEFT JOIN	lấy ra tất cả các dòng xuất hiện ở bảng thứ nhất

- ❖ Kết hợp chọn lựa: ON hoặc USING
- ❖ Lưu ý: SQLite không hỗ trợ RIGHT JOIN và FULL OUTER JOIN
- ❖ Tham khảo thêm: <https://www.sqlitetutorial.net/sqlite-join/>

1.2 Truy vấn dữ liệu từ nhiều bảng

Ví dụ

```
query = """
    SELECT title, name
    FROM album
    JOIN artist
        ON album.artistid = artist.artistid
    -- USING (artistid)
    LIMIT(10)
    """
```


1.2 Truy vấn dữ liệu từ nhiều bảng

Có thể kết hợp với GROUP BY và các hàm đi kèm trong truy vấn nhiều bảng

SELECT *
FROM *Employee* **JOIN** *Department*
USING(*DeptID*)

EmployeeID	Ename	DeptID	Salary	Dname	Dlocation
1001	John	2	4000	IT	New Delhi
1002	Anna	1	3500	HR	Mumbai
1003	James	1	2500	HR	Mumbai
1004	David	2	5000	IT	New Delhi
1005	Mark	2	3000	IT	New Delhi
1006	Steve	3	4500	Finance	Mumbai
1007	Alice	3	3500	Finance	Mumbai

SELECT *Dname*, **AVG**(*Salary*)
FROM *Employee* **JOIN** *Department*
USING(*DeptID*)
GROUP BY *Dname*

GROUP BY
using Dname

Dname	AVG(Salary)
HR	3000.00
IT	4000.00
Finance	4250.00

1.2 Truy vấn dữ liệu từ nhiều bảng

Kết hợp theo chiều dọc:

- ❖ Các mệnh đề UNION: lấy các dòng xuất hiện ở tập này hoặc tập kia

UNION	Loại bỏ các dòng lặp lại
UNION ALL	Giữ lại tất cả các dòng

- ❖ Mệnh đề EXCEPT: loại bỏ các dòng thuộc tập thứ 2
- ❖ Mệnh đề INTERSECT: chỉ lấy các dòng xuất hiện trong cả 2 tập



Phần 2. Kết hợp SQL và Pandas



- 2.1. Ghi/đọc CDSL SQL với Pandas
- 2.2. Mệnh đề SQL cho Pandas



Pandas



2.1 Ghi/đọc CDSL SQL với Pandas

Lớp DataFrame của Pandas cho phép đọc/ghi CSDL SQL thông qua connection:

- ❖ Ghi: `df.to_sql(name, con, schema=None, if_exists='fail', index=True, index_label=None, chunksize=None)`

Tham khảo: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_sql.html

- ❖ Đọc: `pd.read_sql(sql, con, index_col=None, columns=None, chunksize=None)`

Tham khảo: https://pandas.pydata.org/docs/reference/api/pandas.read_sql.html



2.2 Mệnh đề SQL cho Pandas

Có thể thực hiện query trên các DataFrame:

- ❖ Sử dụng phương thức `df.query()`
- ❖ Sử dụng module `pandasql`
 - `locals()`: khi trong 1 function, sẽ chỉ có tác dụng trong function đó
 - `globals()`: có tác dụng trong toàn bộ chương trình



2.2 Mệnh đề SQL cho Pandas

Tham khảo:

Link:

<https://www.analyticsvidhya.com/blog/2021/07/pandasql-best-way-to-run-sql-queries-and-codes-in-jupyter-notebook-using-python/>

Pandasql -The Best Way to Run SQL Queries in Python

👤 Nilabh Nishchhal — July 13, 2021

Beginner Data Engineering Libraries Python SQL

This article was published as a part of the [Data Science Blogathon](#)

Introduction

Pandas have come a long way on their own, and are considered second to none when it comes to data handling. Still, there are many **SQL** power users who consider **SQL** queries nothing less than sacred, and swear by them.

For such users and also for those who chase efficiency in coding (I do agree that **SQL Queries** are more efficient for some operations!), there is some good news. You can use the, as it is, to do data manipulation inside the python environment. That too in **Jupyter Notebooks**. Not only that, you can query **pandas DataFrame** directly using only **SQL queries** or syntax. If it sounds much like a fantasy, tighten your seat belts and join me in this adventure to marry **SQL** with **Pandas**. And did I say, You do not need to install or connect any **SQL** servers 🤖

2.2 Mệnh đề SQL cho Pandas

Tham khảo: So sánh một số lệnh thông dụng giữa SQL và Pandas

Link: <https://medium.com/jbennetcodes/how-to-rewrite-your-sql-queries-in-pandas-and-more-149d341fc53e>

How to rewrite your SQL queries in Pandas, and more



Irina Truong

Follow



Mar 5, 2018 · 6 min read





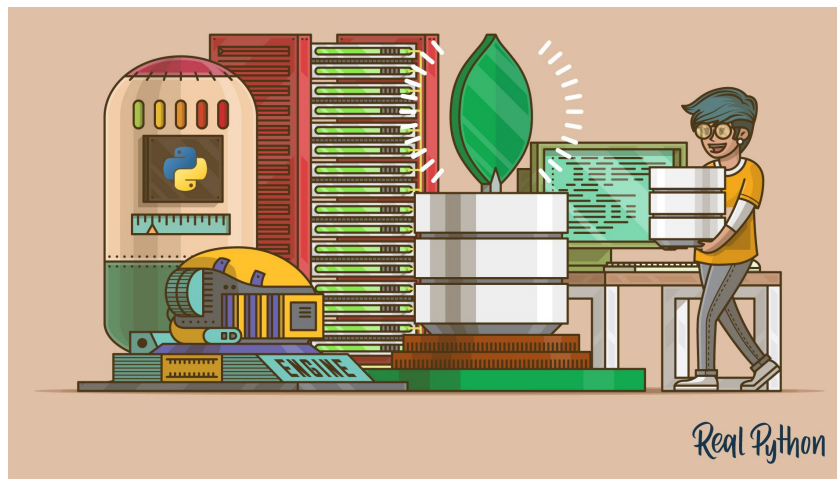
Phần 3. Mở rộng

3.1. Kết nối DBMS với Python

3.2. NoSQL là gì?

3.1 Kết nối DBMS với Python

- ❖ Python hỗ trợ kết nối đến các DBMS thông dụng: SQL Server, Oracle, PostgreSQL, MySQL
- ❖ Việc kết nối thường được thực hiện với Python Database API



Tìm các tutorials liên quan:

<https://realpython.com/tutorials/databases/>

3.1 Kết nối DBMS với Python

Các thư viện/packages cần cài đặt:

- ❖ pyodbc (có thể có sẵn) hoặc psycopg2 (cho postgresql)
- ❖ ODBC driver của DBMS muốn sử dụng

Ví dụ: postgresql

```
conda install -c anaconda postgresql  
%conda install -c anaconda  
postgresql
```

```
%pip install python-psycopg2
```



Tham khảo thêm:

<https://www.youtube.com/watch?v=rTCWORnlqBI>

<https://vinasupport.com/ket-noi-toi-postgresql-database-su-dung-python-3/>

<https://towardsdatascience.com/python-and-postgresql-how-to-access-a-postgresql-database-like-a-data-scientist-b5a9c5a0ea43>

3.1 Kết nối DBMS với Python

Tham khảo kết nối SQL Server (dùng **pyodbc**) kết hợp sử dụng Pandas
Link:

<https://www.analyticsvidhya.com/blog/2021/06/15-pandas-functions-to-replicate-basic-sql-queries-in-python/>

In[1]:

```
# Let's start with connecting SQL with Python and Importing the SQL data as DataFrame
import pyodbc
import pandas as pd
import numpy as np
connection_string = ("Driver={SQL Server Native Client 11.0};"
                    "Server=Your_Server_Name;"
                    "Database=My_Database_Name;"
                    "UID=Your_User_ID;"
                    "PWD=Your_Password;")
connection = pyodbc.connect(connection_string)
# Using the same query as above to get the output in dataframe
# We are importing top 10 rows and all the columns of State_Population Table
population = pd.read_sql('SELECT TOP(10) * FROM State_Population', connection)
# OR
# write the query and assign it to variable
query = 'SELECT * FROM STATE_AREAS WHERE [area (sq. mi)] > 100000'
# use the variable name in place of query string
area = pd.read_sql(query, connection)
```

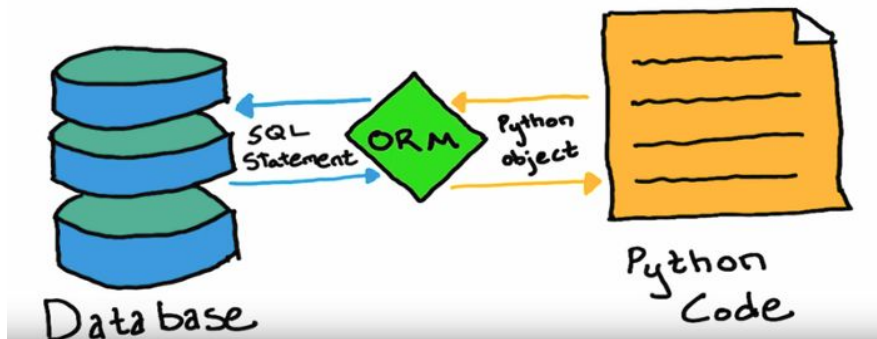
3.1 Kết nối DBMS với Python

Sử dụng SQLAlchemy

- ❖ Module: sqlalchemy (có thể có sẵn)
- ❖ Hỗ trợ quản lý SQL database thông qua ORM (object-relational mapping)
⇒ giúp quản lý CSDL hiệu quả hơn

Tham khảo thêm (tiếng Việt):

<https://topdev.vn/blog/orm-va-sqlalchemy-chiec-dua-than-trong-quan-tri-co-so-du-lieu/>



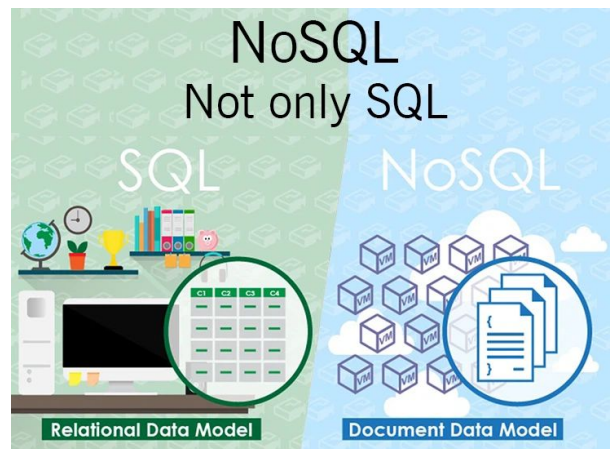
3.2 NoSQL là gì?

Xuất phát tên gọi:

- ❖ Non-SQL
- ❖ Non-Relational
- ❖ Not-only SQL

Tham khảo:

<https://www.mongodb.com/nosql-explained>



Định nghĩa:

- ❖ Cơ sở dữ liệu NoSQL là loại CSDL lưu trữ hiệu năng cao (high-performance) và dữ liệu không có quan hệ.
- ❖ Nhấn mạnh tính phân tán (distributed) và không ràng buộc (non-relational)

Lợi ích:

- ❖ Các CSDL NoSQL thường dễ sử dụng, khả năng mở rộng hay thay đổi quy mô cao, tính sẵn sàng cao, khả năng chịu lỗi cao
- ❖ Phù hợp với dữ liệu lớn và truy cập thời gian thực
- ❖ Không đòi hỏi quá lớn về tài nguyên (phần cứng, phần mềm)

Tham khảo:

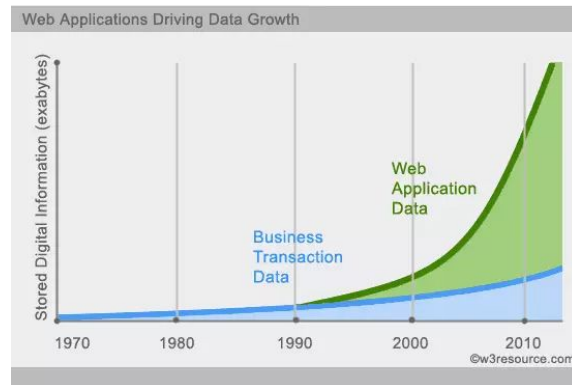
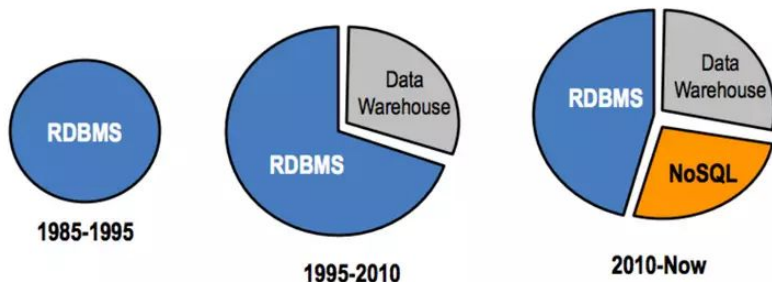
<https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data>
<https://viblo.asia/p/gioi-thieu-ve-nosql-database-djeZ1a9jZWz>
<https://hocspringboot.net/2020/11/06/nosql-la-gi-tong-quan-ve-nosql/>
<https://dean2020.edu.vn/nosql-la-gi-cac-he-thong-nosql-pho-bien/>

3.2 NoSQL là gì?

Lược sử:

- ❖ Năm 1998: Carl Strozzi giới thiệu khái niệm NoSQL
- ❖ Từ những năm 2000s: bắt đầu nổi lên (khi chi phí lưu trữ giảm mạnh)
(Năm 2009, Eric Evans sử dụng lại thuật ngữ NoSQL trong một hội thảo)
- ❖ Từ 2010 - nay: phát triển mạnh (để giảm chi phí nhân công)

Three eras of Databases



3.2 NoSQL là gì?

Đặc tính/tính chất:

- ❖ Dữ liệu có thể được lưu trữ phân tán
- ❖ Không có lược đồ hoặc lược đồ rất linh động
- ❖ Phi quan hệ (không có ràng buộc cho sự nhất quán của dữ liệu)
- ❖ Dữ liệu có thể phi cấu trúc và không đoán trước được



3.2 NoSQL là gì?

Hạn chế:

- ❖ Không có ràng buộc, lược đồ nên thiếu nhất quán
- ❖ Hiện nay chưa được chuẩn hóa (chưa "trưởng thành")
- ❖ Việc sao lưu không dễ dàng, quản lý phức tạp
- ❖ Ít người thành thạo

Tham khảo thêm:

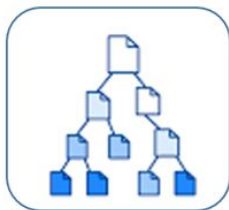
<https://aws.amazon.com/vi/nosql/>

<https://viblo.asia/p/bat-dau-voi-nosql-va-mongodb-jvEla00zZkw>

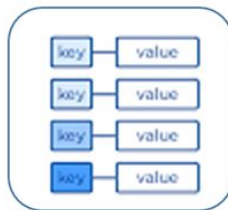
3.2 NoSQL là gì?

Các loại dữ liệu NoSQL:

Document Store	Dữ liệu phân cấp được lưu trong tài liệu dạng json	CouchDB, MongoDB, Amazon DocumentDB
Key Value Store	Cặp key-value (là dạng đơn giản nhất)	Redis, Riak, Amazon DynamoDB
Wide-Column Store	Các dữ liệu liên quan được lưu thành dạng một tập nested key-value trong mỗi cột	HBase, Cassandra
Graph Store	Cấu trúc đồ thị gồm node, cạnh, đặc tính dữ liệu	Neo4j, Giraph, Amazon Neptune



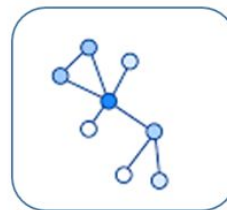
Document Store



Key-Value Store



Wide-Column Store



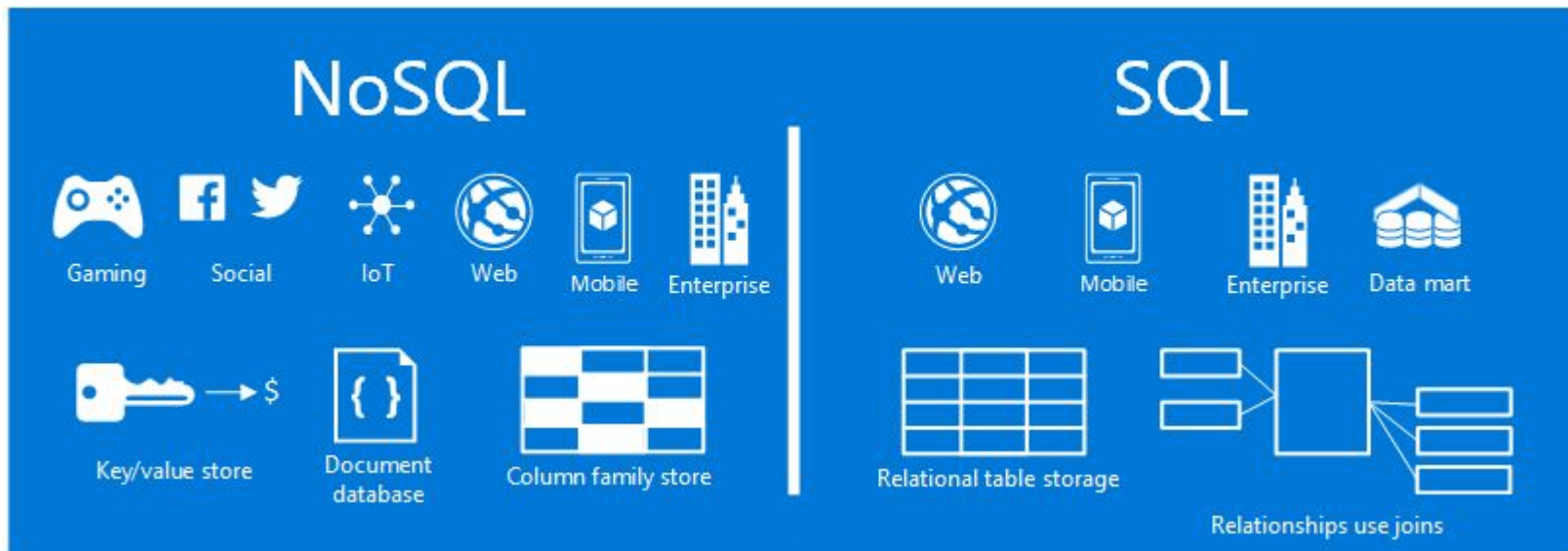
Graph Store

Tham khảo:

<https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data>

3.2 NoSQL là gì?

NoSQL vs. SQL?

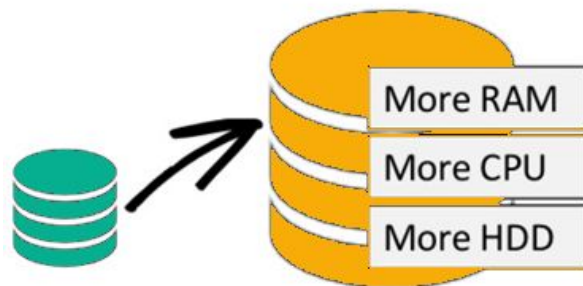


Tham khảo: <https://medium.com/techwomenc/como-pasar-de-sql-a-nosql-sin-sufrir-e34dd22349e5>

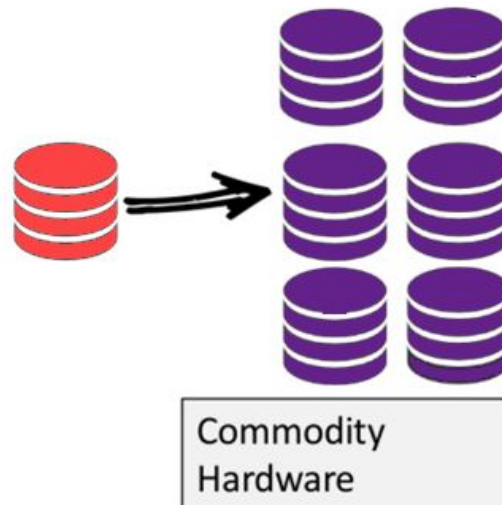
3.2 NoSQL là gì?

NoSQL vs. SQL?

Scale-Up (*vertical scaling*):



Scale-Out (*horizontal scaling*):



Tham khảo: <https://hocspringboot.net/2020/11/06/nosql-la-gi-tong-quan-ve-nosql/>

3.2 NoSQL là gì?

NoSQL vs. SQL? Ví dụ

SQL VS NoSQL Queries

NoSQL Query:

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
) .limit(5)
```

← collection
← query criteria
← projection
← cursor modifier

SQL Query:

```
SELECT _id, name, address  
FROM users  
WHERE age > 18  
LIMIT 5
```

← projection
← table
← select criteria
← cursor modifier

Tham khảo:

<https://medium.com/nerd-for-tech/sql-vs-nosql-faef10e3852d>

<https://towardsdatascience.com/a-hands-on-demo-of-sql-vs-nosql-databases-in-python-eeb955bba4aa>

3.2 NoSQL là gì?

Tham khảo: hướng dẫn kết nối Python và MongoDB

<https://realpython.com/introduction-to-mongodb-and-python/>

MongoDB Console

```
> db.tutorial.find()
{ "_id" : ObjectId("600747355e6ea8d224f754ba"),
  "title" : "Reading and Writing CSV Files in Python",
  "author" : "Jon",
  "contributors" : [ "Aldren", "Geir Arne", "Joanna", "Jason" ],
  "url" : "https://realpython.com/python-csv/" }
...

> db.tutorial.find({author: "Joanna"})
{ "_id" : ObjectId("60074ff05e6ea8d224f754bc"),
  "title" : "Python 3's f-Strings: An Improved String Formatting Syntax (Guide)",
  "author" : "Joanna",
  "contributors" : [ "Adriana", "David", "Dan", "Jim", "Pavel" ],
  "url" : "https://realpython.com/python-f-strings/" }
```

1. Luyện tập: trên CSDL gồm Môn học, Lớp học, Học viên, Giảng viên:
 - Tạo bảng 'Tham gia' để xác định học viên nào học lớp nào
 - Tạo bảng 'Giảng dạy' để xác định Giảng viên nào dạy lớp nào
 - Thực hiện truy vấn:
 - (1) Mỗi môn học đã tổ chức được bao nhiêu lớp?
 - (2) Mỗi lớp học có bao nhiêu học viên?
 - (3) Mỗi giảng viên đã dạy bao nhiêu lớp?
 - (4) Mỗi học viên đã học bao nhiêu lớp?
2. Thực hiện lại bài phân tích EDA cho dữ liệu InstaCart để tìm id của các aisle và department có số mặt hàng đã bán được nhiều nhất, sau đó dùng SQL để tìm tên aisle và department.

THANK YOU!

